AP IFW

Attorney's Docket No.: 10559-299001 / P9310

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant :Victor B. Lortz et al. Art Unit: 2155
Serial No.:09/704,384 Examiner: Kevin T. Bates
Filed :October 31, 2000
Title :NETWORK POLICY DISTRIBUTION

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

Sir:

Applicant herewith files this brief on appeal under 37 CFR 41.37, thereby perfecting the Notice of Appeal that was originally filed on October 12, 2004. The sections required by 37 CFR 41.37 follow.

The appeal brief fee should be charged to deposit account number 06-1050.

(1) Real Party in Interest

The case is assigned of record to Intel Corporation, who is hence presumably the real party in interest.

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

January 12, 2005
Date of Deposit

Signature

Brenda Lewis
Typed or Printed Name of Person Signing Certificate

(2) Related Appeals and Interferences

There are no known related appeals and/or interferences.

(3) Status of Claims

Claims 1-29 are rejected, and all of these claims are reviewed herein.

(4) Status of Amendments

An amendment after final was filed on August 13, 2004. This amendment was considered and apparently will be entered.

(5) Summary of Claimed Subject Matter

Claim 1 requires a method that allows receiving a specification for network policy translation from a first schema to a second schema, translating the network policy into the second schema and configuring a network system based on the translated policy. Page 2 beginning at line 27 describes in general policy server software. Page 3, lines 1-6 describe that the policy includes "a specification 36 for translating the network configuration...". The policy translator 42 can be used to translate the configuration using the translation specification. See for example page 3, lines 9-11. This translated policy is used as part of the network configuration 32.

Claim 9 requires a memory that stores instructions and a processor that carries out similar operations to those discussed above. The operations have been discussed above, and the network and processor are shown in general in figure 1.

The other Independent claims enjoy comparable support within the specification.

(6) Grounds of Rejection

There are two grounds of rejection, one of multiple claims which are rejected under 35 U.S.C. 102 as being anticipated by Eichert. The other ground of rejection is whether other claims are properly rejected as being obvious over Eichert in view of Hind.

(7) Argument

Claims are rejected based on 35 U.S.C. 102 based on Eichert. This contention is respectfully traversed for reasons set forth herein. The rejection states that Eichert receives a specification for translating a network policy, translates the network policy based on the specification and configures the network system. With all due respect, Eichert is an entirely different kind of system, and does not translate the policy in this way and does not meet the limitations of these claims. In fact, Eichert describes the prior art, which is different than the disclosed embodiment, as "policy is represented as flat data, requiring an enforcement device to understand its syntax...". See generally column 4, lines 33-36. This portion goes on to say that if the syntax changes, then all enforcement devices must be reprogrammed to parse the new syntax. Eichert describes using executable code to set the policy. These executable code "objects" form policy definition language. See for example column 3, line 14 through line 32. The objects are created in a ready-to-run state. A system can then run those objects to carry out operations on the network. As described column 3, lines 46-57, the policy is run until the thread finishes its job, at which point the thread is suspended or destroyed. The rejection states that column 10, lines 48-57 meets the claim limitation of translating the policy. However, this cited section describes the use of an executable policy. Admittedly, the policy is described as being in Java and deserialized. The execution and associated with the object is then executed by invoking the method that the object implements. However, this is nearly a byproduct of Eichert's technique of using an executable object as the policy. Even if the deserializing and unwrapping the object could somehow be interpreted as "translating" the policy, it still does not meet the limitations.

Column 10 describes the policy being coded using Java, and admittedly Java can be used on many different processors and operating systems. However, the native Java language, by itself, would be executed by the Java "virtual machine" on any of those processing. This is not "translating," since the same code is run on all processors. Even assuming that this could be considered "translating," there certainly is no teaching of "receiving a specification for translating..." and translating based on that specification which was received, as claimed. At best, the Java code is executed on the Java virtual machine in the each of the different os/processors. There is no teaching of receiving a specification that translates the policy, nor any teaching of actually doing the translation. There is only a teaching of Java, with all that it represents.

Eichert, therefore, does not teach or suggest these features, and therefore does not teach or suggests the subject matter of claim 1, or those claims which depend therefrom. Each of the dependent claims should be similarly allowable. Claim 4, for example, specifies that the file contains both the policy and a specification (which does the translating). Nowhere is this in anyway taught or suggested by Eichert. Eichart simply teach an executable Java object, not a file and a specification for translation.

Claim 5 defines a machine-readable media which includes instructions causing the machine to receive a specification for translating, translate the policy, and configuring network system based on the translated policy. As described above, this is not disclosed by Eichert. Eichert does not teach receiving a specification for translating the policy based on that received specification.

The dependent claims which depend from claim 5 should be similarly allowable.

Claim 9 defines an apparatus which again receives a specification and translates the policy based on that received specification and configures a network system. Claim 9 should therefore be allowable along with the claims that depend therefrom.

Claim 13 defines storing a network policy, storing a specification, translating the network policy and sending the translated network policy to a client computer. As described above, this is not taught or suggested by Eichert, and therefore claim 13 should be allowable for reasons discussed above.

Claim 17 defines an article which includes similar limitations to those discussed above with respect to claim 13 and should be allowable for analogous reasons. Claim 21 defines an apparatus with similar subject matter and should be similarly allowable. Claim 25 specifies transmitting the network policy and a specification for translating the network policy from a server, receiving that, and translating that. This is nowhere taught or suggested by the cited prior art. Claim 27 should be similarly allowable for analogous reasons.

Claims like claim 2 stand rejected as being obvious over Eichert in view of Hind. With all due respect, however, this would be an improper combination. Eichert expressly teaches away from the use of flat data as policy information. In order to obtain the advantages of Eichert, he teaches the use of code, that is executable objects which can be executed to define the network policies.

Hind defines a data policy which is carried out using a style sheet: more specifically "flat data" using the terminology of Eichert. In order to combine Hind with Eichert, therefore, one would have to contradict the express teaching of Eichert not to use flat data. Therefore, one having ordinary skill in the art would not be motivated to make the hypothetical combination of Eichert in view of Hind. Therefore, each of the claims that were rejected over Eichert in view of Hind should be allowable.

In view of the above, therefore, and with all due respect to the patent office's position, the rejection of all of these claims is respectfully traversed. A reversal of the examiner's position is therefore respectfully requested.

Pursuant to 37 DFR §1.136, applicant hereby requests that the period for filing the appeal brief perfecting the Notice of Appeal filed on October 12, 2004 be extended for one month to and including January 12, 2005. Please charge the brief fee and extension fee to Deposit Account No. 06-1050.

Respectfully submitted,

Date: January 12, 2005

Scott C. Harris
Reg. No. 32,030
Attorneys for Intel Corporation

Fish & Richardson P.C.
12390 El Camino Real
San Diego, California 92130
Telephone: (858) 678-5070
Facsimile: (858) 678-5099

10465337.doc

Appendix of Claims

1. A method, comprising:

receiving a specification for translating a network policy from a first schema to a second, different schema;

translating the network policy into the second different schema based on the specification; and

configuring a network system based on the translated policy.

2. The method of claim 1 wherein the network policy is represented in Markup Language which uses tags.

3. The method of claim 1 wherein the specification is received in a file from a policy server.

4. The method of claim 3 wherein the file also contains the policy.

5. An article comprising a machine-readable medium which stores machine-executable instructions for checking events performed by a device, the instructions causing a machine to:

receive a specification for translating a policy from a first schema to a second different schema;

translate the network policy into the second different schema based on the specification; and

configure a network system based on the translated policy.

6. The article of claim 5 wherein the network policy is represented in eXtensible Markup Language and the specification is represented in eXtensible Stylesheet Language.

7. The article of claim 5 wherein the specification is received in a file from a policy server.

8. The article of claim 5 wherein the file also contains the policy.

9. An apparatus comprising:

a memory which stores computer readable instructions; and

a processor which executes the computer readable instructions to:

receive a specification for translating a policy from a first schema to a second, different schema;

translate the network policy into the second different schema based on the specification; and configure a network system based on the translated policy.

10. The apparatus of claim 9 wherein the network policy is represented in eXtensible Markup Language and the specification is represented in eXtensible Stylesheet Language.

11. The apparatus of claim 9 wherein the specification is received in a file from a policy server.

12. The apparatus of claim 9 wherein the file also contains the policy.

13. A method, comprising:

storing a network policy for configuring a network system according to a first schema;

storing a specification for translating the network policy from the first schema to a second different schema;

translating the network policy into the second different schema based on the specification; and

sending the translated network policy to a client computer.

14. The method of claim 13, further comprising:

prior to translating the network policy:

sending the network policy to the client computer;

sending the specification for translating the network policy to the client computer; and

receiving an indication that the client computer cannot translate the network policy.

15. The method of claim 13 wherein the network policy is represented in eXtensible Markup Language and the specification is represented in eXtensible Stylesheet Language.

16. The method of claim 13 wherein the network policy and the specification are stored in one file.

17. An article comprising a computer-readable medium which stores computer-executable instructions for checking events performed by a device, the instructions causing a machine to:

store a network policy for configuring a network system according to a first schema;

store a specification for translating the network policy from the first schema to a second different schema;

translate the network policy into the second different schema based on the specification; and send the translated network policy to a client computer.

18. The article of claim 17, wherein the instructions further cause the machine to:

prior to translating the network policy:

send the network policy to the client computer;

send the specification for translating the network policy to the client computer; and

receive an indication that the client computer cannot translate the network policy.

19. The article of claim 17 wherein the network policy is represented in eXtensible Markup Language and the specification is represented in eXtensible Stylesheet Language.

20. The article of claim 17 wherein the network policy and the specification are stored in one file.

21. An apparatus comprising:

a memory which stores computer readable instructions;

a processor which executes the computer readable instructions to:

store a network policy for configuring a network system according to a first schema;

store a specification for translating the network policy from the first schema to a second different schema;

translate the network policy into the second different schema based on the specification; and

send the translated network policy to a client computer.

22. The apparatus of claim 21 wherein, prior to translating the network policy, the processor executes the instructions to:

send the network policy to the client computer;

send the specification for translating the network policy to the client computer; and

receive an indication that the client computer cannot translate the network policy.

23. The apparatus of claim 21 wherein the network policy is represented in eXtensible Markup Language and the specification is represented in eXtensible Stylesheet Language.

24. The apparatus of claim 21 wherein the network policy and the specification are stored in one file.

25. A method of configuring a network comprising:

transmitting a network policy according to a first schema and a specification for translating the network policy from the first schema to a second different schema from a server;

receiving the network policy and the specification on a first client computer;

translating on the client computer the network policy from the first schema to the second different schema using the specification; and

configuring the network system on the first client computer using on the translated network policy.

26. The method of claim 25 further comprising:

receiving the network policy on a second client computer; and

configuring the network system on the second client computer using on the network policy.

27. The method of claim 25 further comprising:

receiving the network policy on a third client computer;

transmitting to the server an indication that the third client computer cannot translate the network policy;

translating on the server the network policy from the first schema to the second different schema using the specification; and

transmitting the translated network policy to the third client computer.

28. The method of claim 27 wherein the network policy is represented in eXtensible Markup Language and the specification is represented in eXtensible Stylesheet Language.

29. The method of claim 27 wherein the network policy and the specification are contained in one file.